Loudini et al. Journal of Internet Services and Applications 2013, 4:15
http://www.jisajournal.com/content/4/1/15

Journal of Internet Services and Applications
a SpringerOpen Journal

RESEARCH Open Access

Incorporate intelligence into the differentiated services strategies of a Web server: an advanced feedback control approach

Malik Loudini*, Sawsen Rezig and Yahia Salhi

Abstract

This paper presents an investigation into the application of advanced feedback control strategies to provide better web servers quality of service (QoS). Based on differentiated service strategies, fuzzy logic based control architectures are proposed to enhance the system capabilities. As a first control scheme, a Mamdani fuzzy logic controller (FLC) is adopted. Then, the Simulated Annealing (SA) algorithm (SAA) is used to optimize the FLC parameters with efficient tuning procedures. The SA optimized FLC (SAOFLC) is also implemented and applied to improve the system QoS. Simulation experiments are carried out to examine the performances of the proposed intelligent control strategies.

Keywords: Web server, Quality of service, DiffServ, Service delay guarantee, Absolute delay, Relative delay, Fuzzy logic controller, Simulated annealing

1 Background

With the tremendous growth of internet and its extraordinary success, the web servers become more and more numerous and diverse. They are, also, more and more exposed to high rates of incoming requests from users which are becoming increasingly reliant on these new sorts of modern service delivery. Providing high dynamic contents, integrating with huge databases and offering all sorts of complex and secure transactions, these internet applications are faced with growing difficulties to ensure adequate QoS [1].

Evaluation of web server QoS performance generally focuses on achievable delay of service or response time for a request-based type of workload as a function of a traffic load.

Adopting such metrics, many QoS performance enhancement architectures and mechanisms, particularly based on differentiation of service (DiffServ) [1-3], have been proposed by the community of researchers in this area. Among these, the feedback control (or closed-loop control) has been occupying a place of predilection.

Indeed, applying feedback control schemes to enhance the performance of software processes is becoming an attractive research area. The main advantage offered by this technique of automatic control is its robustness to modeling inaccuracies, system nonlinearities, and time variation of system parameters. These types of uncertainties are very common in unpredictable poorly modeled environments such as the Internet. For a literature review about the application of feedback control to computing systems, see [4-7].

Most of the feedback control techniques and algorithms are relying on the availability of formal parametric models of the controlled system and control theoretic tools. This is not always possible for software processes for which analytical models are not easily obtainable or the models themselves, if available, are too complex and nonlinear.

Furthermore, it is well known that web workloads are stochastic with significant parameter variations over time. So, a challenging problem is how to provide efficient performance control over a wide range of workload conditions knowing the highly nonlinear behavior of a web server in its response to the allocated resources.

It is precisely for processes and environments such these that we need judicious non-conventional control

* Correspondence: m_loudini@esi.dz
Ecole Nationale Supérieure d'Informatique (ESI), Laboratoire de Communication dans les Systèmes Informatiques (LCSI), B.P 68M, 16270 Oued Smar, El Harrach, Algiers, Algeria

algorithms that will be implemented without dependency on the availability of the above-mentioned requirements.

Computational intelligent approaches to handle the complexity and fuzziness present in such software systems surely have an essential role to play. We should therefore exploit their tolerance for imprecision and uncertainty to achieve tractability and robustness in control applications.

Feedback control schemes based on Fuzzy Logic Controllers (FLCs) are well known for their ability to adapt to dynamic imprecise and bursty environments such that of the web traffic.

It appears that this category of intelligent control structures should therefore be the most recommended.

In this paper, web server QoS enhancement solutions based on closed-loop intelligent control strategies, including fuzzy logic, are investigated.

As related works to our study context, examples of earlier relevant research investigations, using various control techniques, can be found in [8-23].

The remainder of this paper is organized as follows. In Sect. 2, we briefly describe how web servers operate, then we present some semantics of delays and service delay guarantees in web servers. We also briefly call back the main basics about fuzzy control. An introduction to the SA optimization method is given at the end of the section. In Sect. 3, the modeling of the web server system is described and different discrete models are given. In Sect. 4, we present the adopted feedback control strategy aimed to satisfy the desired performance of the web server. The implementation details and the simulation results are given in Sect. 5. Section 6 presents the related work. Finally, Sect. 7 concludes the paper.

2 Preliminaries

In this section, we briefly describe how web servers operate and then present some semantics about delays and service delay guarantees. We also briefly call back the main basics about fuzzy control and introduce the SA optimization method.

2.1 Web servers

Web servers are commonly defined as computers that deliver web pages. Having an IP address and generally a domain name, a web server is software responsible for accepting HTTP [24] requests from clients and offering them services as HTTP responses. HTTP lies behind every web transaction. An HTTP transaction consists of three steps: TCP [25] connection setup, HTTP layer processing and network processing. Once the connection has been established, the client sends a request for an object (HTML file, image file …). The server handles the request and returns the object of this query [26].

It is well known that web servers adopt either a multi-threaded or a multi-process model to handle a large number of users simultaneously. Processes or threads can be either created on demand or maintained in a pre-existing pool that awaits incoming TCP connection requests to the server. In HTTP 1.0, each TCP connection carried a single web request. This resulted in an excessive number of concurrent TCP connections. To remedy this problem the new version of HTTP, called HTTP 1.1 [27], reduces the number of concurrent TCP connections with a mechanism called *persistent connections*, which allows multiple web requests to reuse the same connection [8].

As in [8,13], a multi-process model with a pool of processes is assumed, which is the model of the Apache server, the most commonly used web server today [28].

2.2 Differentiation of services

Differentiated Services (commonly known as DiffServ) has been proposed by the IETF Differentiated Services Working Group [2]. It is a computer networking protocol or architecture that allows different levels of services on a common network in order to provide a better QoS. In other words, it supports a manageable and scalable service differentiation for class-based aggregated traffic in IP networks. Two approaches exist in DiffServ architecture:

Absolute DiffServ: This model seeks to guarantee end-to-end QoS. In this architecture, the user receives an absolute service profile (e.g., end-to-end delay or bandwidth guarantee …) and the network administrator attempts to maintain the absolute metric spacing between the users classes.

Relative DiffServ: This model seeks to provide relative or proportional services. In other words, it aims to guarantee to a higher priority class of users better (proportionally ratioed) service performances than those provided to a lower priority class.

2.3 Service delay guarantees: semantics, definitions and adopted Qos metrics

Our investigation being concerned with delays based QoS enhancement, we begin this paragraph by giving useful semantics and definitions relative to the service delay differentiation approach [13].

First, every HTTP request being supposed to belong to a class k ($0 \leq k < N$), two main delays are defined as:

Processing delay: It is the time interval between the arrival of an HTTP request to the process responsible for the corresponding connection and time the server completes transferring the response.

Connection delay: It is the time interval between the arrival of a TCP connection (establishment) request and the time where the connection is accepted (dequeued)

by a server process. The connection delay includes the queuing delay. In other words, the connection delay of class k at the m^{th} sampling instant, denoted by $C_k(m)$, is defined as the average connection delay of all established connections of class k within the time interval $[(m-1)T_s, mT_s]$, where T_s is a constant sampling period.

The delay differentiation being applied to connection delays, the adopted QoS metrics in this work are the connection delay guarantees. Using, for simplicity, delay to refer to connection delay, they are defined as follows:

Relative delay guarantee: A desired relative delay (RD) W_k is assigned to each class k. A RD guarantee $\{W_k | 0 \le k < N\}$ requires that $C_j(m)/C_l(m) = W_j(m)/W_l(m)$ for classes j and l ($j \neq l$).

Absolute Delay Guarantee: A desired absolute delay (AD) W_k is assigned to each class k. An AD guarantee $\{W_k | 0 \le k < N\}$ requires that $C_j(m) \le W_j(m)$ for any class j if there exists a lower priority class $l > j$ and $C_l(m) \le W_l(m)$ (a lower class number means a higher priority). Note that since system load can grow arbitrarily high in a web server, it is impossible to satisfy the desired delay of all service classes under overload conditions. The AD guarantee requires that all classes receive satisfactory delay if the server is not overloaded; otherwise desired delays are violated in the predefined priority order, i.e., low priority classes always suffer guarantee violation earlier than high priority classes.

2.4 Brief review of fuzzy control

The structure of a process controlled via a Mamdani type FLC [29,30] is shown in Figure 1.

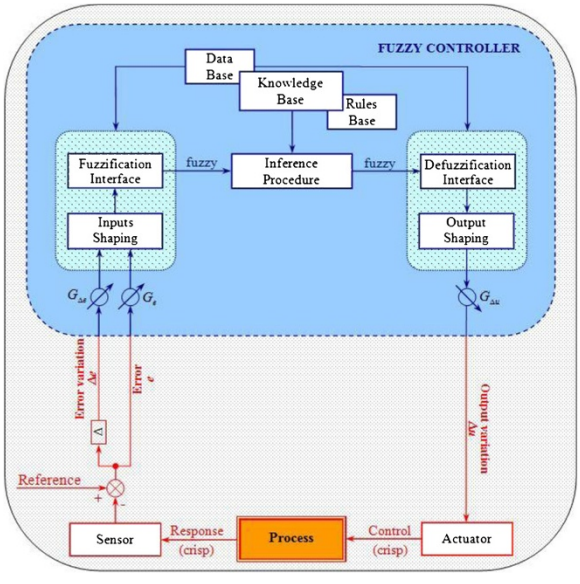


Figure 1 Basic structure of a process controlled via a Mamdani FLC.

The basic components of the considered FLC are briefly presented below:

- The *fuzzification interface* gets the values of input variables (e, Δe), performs a scale mapping to transfer the range of their values into corresponding universes of discourse, and performs the function of fuzzification to convert input (crisp) data into linguistic values.
- The *knowledge base* comprises a rule base which characterizes the control policy and goals.
- The *data base* provides the necessary definitions about discretization and normalization of universes, fuzzy partition of input and output spaces, membership functions (MFs) definitions.
- The *inference procedure* process fuzzy input data and rules to infer fuzzy control actions employing fuzzy implication and the rules of inference in fuzzy logic.
- The *defuzzification interface* performs a scale mapping to convert the range of values of universes into corresponding output variables, and transformation of a fuzzy control action inferred into a nonfuzzy control action (Δu).
- G_e, $G_{\Delta e}$ are the inputs scaling factors and $G_{\Delta u}$ is the output scaling factor.

2.5 Simulated annealing

Inspired from nature, simulated annealing (SA) is a powerful stochastic local search algorithm first introduced by Metropolis et al. [31] as a modified Monte Carlo integration method and then proposed and made popular by Kirkpatrick et al. [32] to solve difficult combinatorial optimization problems. SA is based on the analogy between the annealing of solids and the solving of combinatorial optimization problems. Annealing is the process through which a solid material is initially heated over the melting point to be liquefied with randomly dispersed particles. Then the material is cooled slowly until it crystallizes into a state of perfect lattice according to a cooling scheduled.

3 Web server dynamic modeling

The systematic design of feedback systems requires an ability to quantify the effect of control inputs (e.g., buffer size) on measured outputs (e.g., response times), both of which may vary with time. Indeed, developing such models is at the heart of applying control theory in practice [5]. The models obtained are also used to make numerical simulations as needed in this work.

Our control investigation will be tested based on the dynamic models established in [13]. The approach employed, in deriving the mathematical models, is

statistical (black-box method), a process that is referred to as system identification [33].

The system to be controlled is modeled as a difference equation with unknown parameters.

The web server is stimulated with pseudo-random digital white-noise input and a least squares estimator [33] is used to estimate the model parameters.

The details about the conducted experiments and the obtained results can be found in [13]. Lu et al. have established that, for both RD and AD control, the controlled system can be modeled as a second order difference equation with adequate accuracy for the purpose of control design. A brief presentation is given below.

The web server is modeled as a difference equation with unknown parameters, i.e., a n the order model can be described as follows:

$$V(m) = \sum_{j=1}^{n} a_j V(m-j) + \sum_{j=1}^{n} b_j U(m-j) \quad (1)$$

In a n the order model, there are $2n$ parameters $\{a_j, b_j |\ 1 \le j < n\}$ that need to be decided by the least squares estimator.

The system identification results established that, the controlled system can be modeled by the following second order difference equation:

$$V(m) - a_1 V(m-1) - a_2 V(m-2) = b_1 U(m-1) + b_2 U(m-2) \quad (2)$$

The system model defined by the difference equation (2) can be, easily, converted to a description by a discrete transfer function $G(z)$ from the control input $U(z)$ to the output $V(z)$ in the z -domain, given below:

$$G(z) = \frac{V(z)}{U(z)} = \frac{b_1 z + b_2}{z^2 - a_1 z - a_2} \quad (3)$$

The stimulation of the web server being carried out based on SURGE [34] as the HTTP requests generator, two sets of experiments has been conducted, using three workloads with different user populations, for each of the two adopted approaches in service differentiation: the RD case and the AD case (see Table 1).

The variation of user populations (2 classes) is aimed to evaluate the sensitivity of the model parameters to workloads.

For each experience, a difference equation based dynamic model has been established. The resulting discrete transfer functions are given in Table 1.

4 Design of the FLC based feedback control system

In this section, we first present the global feedback control architecture for web server QoS, and then formally specify the proposed controllers.

4.1 Global feedback control architecture

The adopted feedback control architecture is illustrated in Figure 2.

In this architecture, the controlled system is the web server. The connection scheduler serves as an actuator transmitting, at each sampling instant m, the control input effort in terms of *process budgets* $\{B_k |\ 0 \le k < N\}$ (input U) computed and generated by the controller based on the errors provided by the feedback loops. These errors result from the comparisons between the desired relative or absolute delays $\{W_k |\ 0 \le k < N\}$ and the measured delays or the sampled connection delays $\{C_k |\ 0 \le k < N\}$ (output V) computed by the monitor at each sampling instant. For each of the AD and RD approaches, the control key variables are explicitly summarized in Table 2.

4.2 Derivation of the FLC

The FLC based web server process control strategy adopted in our work is illustrated in Figure 3.

This scheme, by its structure, is also called "Mamdani PI type FLC" where PI stands for Proportional-Integral.

The input variables of the FLC are the loop error e and its rate of change Δe which are defined as:

$$e(mT_s) = Ref(mT_s) - WSR(mT_s) \quad (4)$$

where Ref is the reference input, WSR is the web server response, and mT^s is a sampling interval,

$$\Delta e(mT_s) = \frac{\{e(mT_s) - e[(m-1)T_s]\}}{T_s} \quad (5)$$

Table 1 Experiments data and corresponding transfer functions

	RD case			AD case		
	Class 0	Class 1	Transfer function $G(z)$	Class 0	Class 1	Transfer function $G(z)$
Workload A	200	200	$\frac{0.95z-0.12}{z^2-0.74z+0.37}$	100	400	$\frac{-0.82z-0.52}{z^2+0.13z+0.03}$
Workload B	150	250	$\frac{2.28z+0.08}{z^2-0.31z+0.27}$	150	250	$\frac{-0.36z-0.15}{z^2-0.14z+0.05}$
Workload C	300	300	$\frac{0.47z+0.21}{z^2-0.56z+0.26}$	200	300	$\frac{-0.49z-0.25}{z^2-0.25z+0.03}$

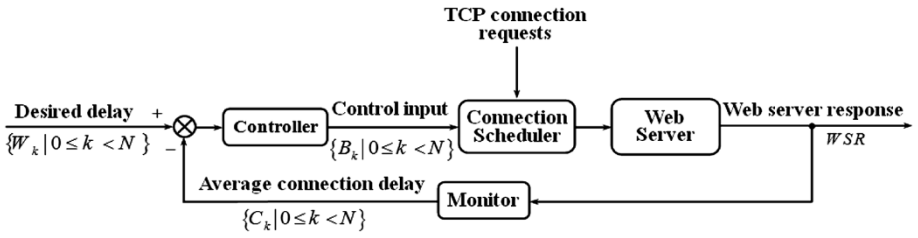


Figure 2 Web server feedback control general scheme.

The change in the control setting is denoted by $\Delta u(mT_s)$. G_e, $G_{\Delta e}$ are the inputs scaling factors and $G_{\Delta u}$ is the output scaling factor. Thus the PI type fuzzy logic command is given by

$$u(mT_s) = u[(m-1)T_s] + G_{\Delta u} * \Delta u(mT_s) \quad (6)$$

2.3 Derivation of the SAOFLC

In order to try to improve the performances of the previous FLC designed based on observations and subjective choices, we apply the SA as an optimization algorithm to automatically adjust its design parameters:

- Number of MFs for each FLC variable
- MFs shapes for each FLC variable
- MFs distribution for each FLC variable
- Decision table rules
- Scaling factors.

The SAA tuning procedure is carried out according to the pseudo code provided in Figure 4.

4.3.1 Conception hypotheses and constraints

Certain assumptions and constraints about the decision table and the FLC variables MFs to be optimized are given here:

- The number of fuzzy sets (NFS) for each variable can take only one of the following possible values: 3, 5, 7 or 9.
- The fuzzy sets (FSs) will be symbolized (labeled) by the standard linguistic designation and indexed by an ascending order. If, for example, the number of FSs of a linguistic variable is equal to 5, the corresponding FSs will be: NB, NM, ZE, PM, PB and indexed from 1 to 5. The FSs NB and NM are considered as the opposites to PB et PM respectively (symmetrically with respect to ZE).
- Note that the label ZE stands for linguistic (fuzzy) value zero, first letters N and P mean negative and positive and second letters B, M and S denote big, medium and small values respectively.
- All the FLC variables universes of discourse are normalized to lie between −1 and +1.
- The first and the last MFs have their apexes at −1 and +1 respectively.

4.3.2 Decision rules table deriving method

The adopted method for the decision rules table construction is inspired from the works developed in [35,36].

As a contribution, a new method of FSs assignment to each of the grid nodes in the special case of equality of distances between the points representing the candidate decision rules is proposed (see the decision rules table deriving method principle given below).

Note that this new procedure is adopted instead of the random assignment proposed in [36].

Principle of the method First, the grid is constructed using two spacing parameters PSG_e and $PSG_{\Delta e}$ relatively to the FLC two inputs e and Δe.

The first (resp. the second) spacing parameter $PSGe$ (resp. $PSG_{\Delta e}$) fix the grid nodes X-axis coordinates (resp. Y-axis coordinates) in the interval [−1, +1] (universe of discourse (UD)) with a simple computing formula given in the next paragraph. Each abscissa (resp. ordinate) represents a fuzzy set (FS) of the variable e (resp. Δe). The number of the grid constitutive nodes is then equal to the product result between the two FLC input FSs

Table 2 Variables of the feedback control scheme

	AD	RD
Reference W_k	Desired delay of class k	Desired delay ratio between class k and k - 1
Output C_k (V)	Measured delay of class k	Measured delay ratio between class k and k - 1
Control input B_k (U)	Process budget of class k	Ratio between the process budgets of classes k and k

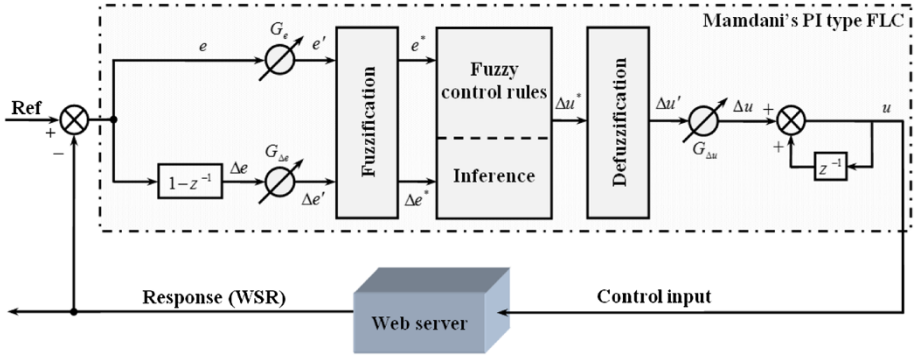


Figure 3 FLC based web server feedback control scheme.

numbers. Once, the nodes are fixed, we introduce the output points on a straight line corresponding to the FLC output variable Δu. Now, the points (output ones) represent the FSs and not their coordinates. The number of points is equal to the output variable FSs number.

A third spacing parameter $PSG_{\Delta u}$ fix the output points X-axis (Y-axis) coordinates similarly with the nodes fixing manner whereas the Y-axis (X-axis) coordinates are calculated by an angular parameter, noted "*Angle*", which determine the slope of the straight line, supporting the output points, with respect to the horizontal. This angular parameter varies in the interval $[0, \pi/2]$ counterclockwise.

Each of the grid nodes represents a case of the decision table and each output point represents a FS of the control variable Δu.

Pseudo code of the SAA

Get an initial FLC FC_0 /*Initial solution*/
Let $FC = FC_0$
Let $Of = Of\ (FC)$ /* Of () : Objective function*/
Set an initial temperature T and a final temperature T_{fin}
While $(T > T_{fin})$
  While ($Nit > 0$) /* Nit : Number of iterations*/
    Generate the neighborhood $h\ (FC)$
    $FC_n \leftarrow FC_h$ /* FC_h is randomly selected in $h\ (FC)$ */
    If $Of\ (FC_n) - Of\ (FC) \leq 0$ **then** $FC \leftarrow FC_n$
      Else
    Generate a random number Rnb between 0 and 1
    If $Rnb < e^{[Of\ (FC_n) - Of(FC)]\ /T}$ **then** $FC \leftarrow FC_n$
    Endif
  End While
  Generate a random real α in [0,1] (Cooling Coefficient)
  $T \leftarrow T * \alpha$
End While

Figure 4 Pseudo code of the simulated annealing algorithm.

Once all the points coordinates (grid nodes and output points) are computed, we can proceed to the assignment by determining the minimal distance among all the distances separating each node of the grid from all the output points situated on the straight line. Then, we assign to each node of the grid the closest output point. Consequently, the decision table case corresponding to this node will contain the FS representing the selected output point. Nevertheless, an assignment conflict could arise in the case of equality between two minimal distances separating a node and two output points. We have proposed to select the output point which has the lower FS index if it is a case of the upper part with respect to the table diagonal or the output point which has the greater FS index if the case belongs to the lower part [37]. It should be noted that no more than two output points can be at the same distance from a given node of the grid since all the output points are on the same straight line.

Spacing parameter The grid spacing parameter *PSG* specifies how the positions C_1 of the intermediate points (between the center and the extreme of each graduated axis) are spaced out with respect to the central point.

This parameter offers flexibility in varying spacing. The more it is greater than 1, the more the points positions are closest to centre and vice versa. At the value 1, the positions are uniformly distributed in the UD interval [−1, 1].

The number of positions C_1 and FSs being obviously the same, we have proposed a formulation of the spacing law in function of the spacing parameter *PSG* [37,38].

At a first stage, the positions C_i being equidistant are denoted by CEq_i and computed by:

$$CEq_i = 2\left(\frac{i-1}{NEF-1}\right) - 1, i = 1, \cdots NFS \tag{7}$$

The C_i values are, then, determined in terms of the spacing parameter *PSG* as follows:

$$C_i = sign(CEq_i) * |CEq_i|^{PSG} \quad (8)$$

with $sign(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$; $PSG = (PSG_1)^{PSG_2}$ with PSG_2 that can take the values +1 or −1.

Two illustrative examples of C_i computation are given in Table 3 for 7 FSs and 5 FSs, respectively, and for different values of the spacing parameter.

To understand the decision table deriving procedure, two detailed examples are given bellow. The constructing parameters are given in Table 4, then, the grids and their corresponding decision tables are shown in Figures 5 and 6 respectively.

Note that the nodes are represented by red stars and the output points by blue circles. The purple arrows are examples of minimal distances between the output points and the grid nodes describing the FSs assignment to the decision table.

It is interesting to note that the decision table obtained for $PSG_e = PSG_{\Delta e} = PSG_{\Delta u} = 1$ and $Angle = 45°$ is none other than the Mac Vicar-Whelan diagonal table [39].

4.3.3 Membership functions deriving method

Determination of the FLC MFs using the SAA takes place in three phases:

1. creation of primary MFs of the FLC input/output parameters,
2. parameterization,
3. adjustment of the MFs.

MFs shape and width optimization

Three types of MFs shapes are considered:

- triangular
- trapezoidal which include (generalize) the triangular one
- "two-sided" Gaussian with flattened summit

Table 3 C_i in function of *PSG* for 7 FSs

	PSG	*Ci*						
		C1	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*
Example 1	0.25	−1	−0.90	−0.76	0	0.76	0.90	1
	0.5	−1	−0.81	−0.58	0	0.58	0.81	1
	1	−1	−0.67	−0.33	0	0.33	0.67	1
	2	−1	−0.44	−0.11	0	0.11	0.44	1
	4	−1	−0.20	−0.01	0	0.01	0.2	1
Example 2	0.25	−1	−0.84	0	0.84	1		
	0.5	−1	−0.70	0	0.70	1		
	1	−1	−0.50	0	0.50	1		
	2	−1	−0.25	0	0.25	1		
	4	−1	−0.06	0	0.06	1		

Table 4 Two illustrative examples

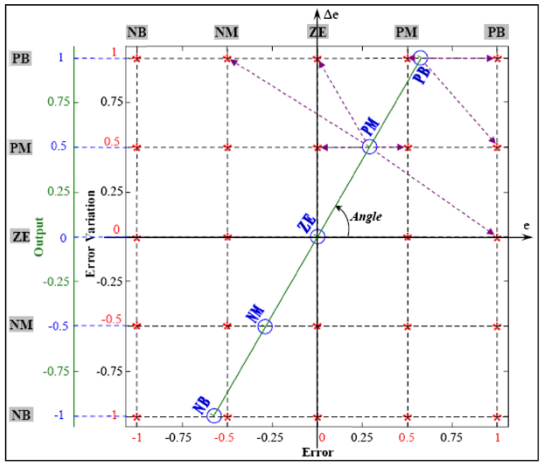
	NFS_e	$NFS_{\Delta e}$	$NFS_{\Delta u}$	PSG_e	$PSG_{\Delta e}$	$PSG_{\Delta u}$	*Angle*
Example 1	5	5	5	1	1	1	60°
Example 2	5	5	5	0.5	1	2	30°

The triangular shape is defined by three parameters $[P1 \quad P2 \quad P3]$ which represent respectively, the left abscissa of the triangle base, the peak abscissa, and the right abscissa of the triangle base.

Each triangle base begins at the precedent triangle peak abscissa and ends at that of the following one. The trapezoidal shape is defined by four parameters $[P1 \quad P2 \quad P3 \quad P4]$ representing, respectively, the base left abscissa, the summit left abscissa, the summit right abscissa, and the base right abscissa.

The trapezoidal shape is then framed by four points with the coordinates: $(P1, 0)$, $(P2, 1)$, $(P3, 1)$ and $(P4, 0)$. Note that if $P2 = P3$, we obtain a triangular shape (see Figure 7).

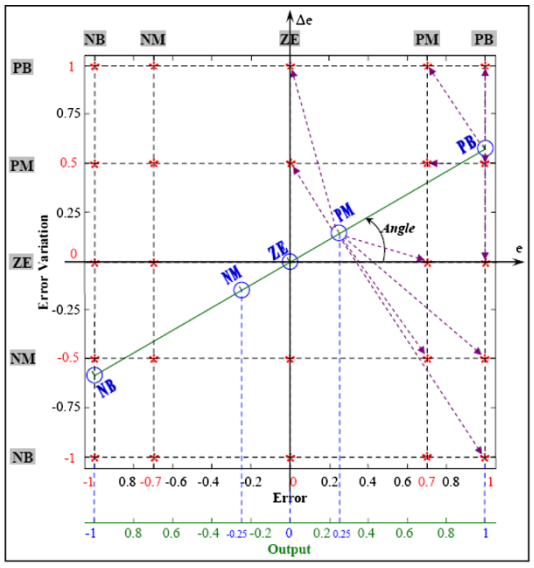
We also define the two-sided Gaussian shape by four parameters $[Sig1 \quad G1 \quad G2 \quad Sig2]$ (see Figure 8). The



Δu	e				
	NB	NM	ZE	PM	PB
NB	NB	NB	NM	NM	ZE
NM	NB	NM	NM	ZE	ZE
Δe ZE	NM	ZE	ZE	ZE	PM
PM	ZE	ZE	PM	PM	PB
PB	ZE	PM	PM	PB	PB

Figure 5 Example 1 grid and corresponding decision table: (a) Grid constitution; (b) Derived decision table.

a



b

Δu		e				
		NB	NM	ZE	PM	PB
Δe	NB	NB	NB	NM	ZE	PM
	NM	NB	NB	NM	PM	PM
	ZE	NB	NM	ZE	PM	PB
	PM	NM	NM	PM	PB	PB
	PB	NM	ZE	PM	PB	PB

Figure 6 Example 2 grid and corresponding decision table: (**a**) **Grid constitution**; (**b**) **Derived decision table.**

left and right sides of the Gaussian are respectively defined by: $G(x) = e^{-\frac{(x-G1)^2}{2(Sig1)^2}}$ and $G(x) = e^{-\frac{(x-G2)^2}{2(Sig2)^2}}$.

To be able to use this two-sided Gaussian shape within the framework of our optimizing method, we must bound this shape by the same points used for the trapezoidal shape (Figure 7). In other words, we must define the two-sides Gaussian shape in terms of the parameters $[P1 \quad P2 \quad P3 \quad P4]$ instead of $[Sig1 \quad G1 \quad G2 \quad Sig2]$. For that purpose, we adopted a very small positive real number ε ($\varepsilon = 0.01$ was quite suitable) such that:

- The Gaussian left curve includes the points *(P1,ε)* and *(P2,1)*.
- The Gaussian right curve includes the points *(P3,1)* and *(P4,ε)*.

This formulation leads to the establishing of the following two systems of equations:

$$\begin{cases} e^{-\frac{(P1-G1)^2}{2*(Sig1)^2}} = \varepsilon \\ e^{-\frac{(P2-G1)^2}{2*(Sig1)^2}} = 1 \end{cases} \tag{9}$$

$$\begin{cases} e^{-\frac{(P3-G2)^2}{2*(Sig2)^2}} = 1 \\ e^{-\frac{(P4-G2)^2}{2*(Sig2)^2}} = \varepsilon \end{cases} \tag{10}$$

The resolution of systems (9) and (10) gives:

$$G1 = \mathrm{P2}; G2 = \mathrm{P3}; Sig1 = \sqrt{-\frac{(\mathrm{P1}-\mathrm{P2})^2}{2*\log\varepsilon}};$$

$$Sig2 = \sqrt{-\frac{(\mathrm{P4}-\mathrm{P3})^2}{2*\log\varepsilon}}.$$

Note that ε has been used since the Gaussian two sides never pass by a null abscissa.

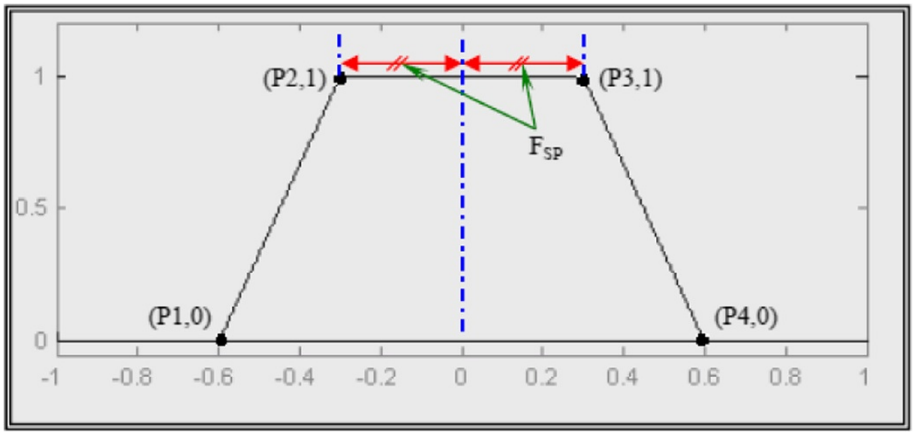


Figure 7 Trapezoidal MF.

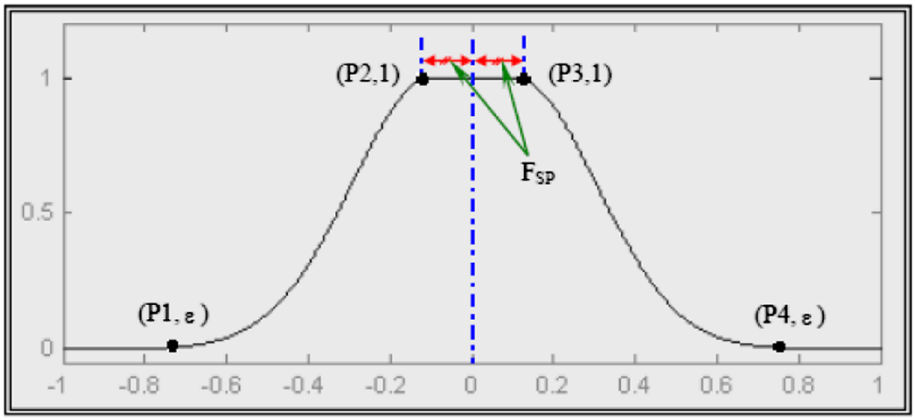


Figure 8 Two-sided Gaussian MF.

Width spacing parameter The summit abscissae of the different shapes are calculated with the same principle of parameter spacing used in the determination of the grid nodes and the points coordinates in the decision table derivation. The FLC input/output variables MFs spacing parameters are, respectively, denoted by PSF_e $PSF_{\Delta e}$ and $PSF_{\Delta u}$.

Shape optimizing parameter The MFs spacing method being inspired by the works of Park et al. [35], Foran [36], and Cheong and Lai [40], we propose a new technique for the MFs shape optimization [37] based on a design parameter called shape parameter (*SP*). This optimizing parameter gives possibilities of diversification (hybridization) of MFs shapes on the UD of each of the FLC input/output variables.

SP is considered as a real number belonging to the interval [0, 2[. Its integer part, denoted by I_{SP}, will determine the shape of the MFs and its fractional one, denoted by F_{SP}, will determine the spacing with respect to the center of the MF. The MF shape is specified by I_{SP} and F_{SP} as follows:

- $I_{SP} = 0$: trapezoidal or triangular shape
- $I_{SP} = 1$: two-sided Gaussian shape.
- F_{SP} determines the symmetric space with respect to the center of the MF as shown in Figure 7 and Figure 8. As we can see in Figure 7, if the spacing is equal to zero, the trapezoidal shape reduces to a triangular one.

Being optimized by the SAA, the number of MFs (*NFS*) for each of the FLC input/output variables, is not constant. Consequently, it is not feasible to assign a spacing parameter to each MF. So, we propose a solution, which consists in allocating a shaping parameter, denoted by SP_M, for the MF of the middle of the UD and another, denoted by SP_E, for the extreme MF.

The intermediate MFs shaping parameters, denoted by SP_I, are then deducted from SP_M and SP_E so that they will have equidistant intermediate values.

The i^{th} shape parameter $SP_I(i)$ corresponding to the i^{th} intermediate MF, is determined by:

$$SP_I(i) = SP_M + 2(i-1)\frac{SP_E - SP_M}{NFS-1}; \quad i = 1, ..., \frac{NFS+1}{2} \tag{11}$$

We can observe that $SP_I(1) = SP_M$ and $SP_I\left(\frac{NFS+1}{2}\right) = SP_E$. So, two parameters are enough for any number of FSs.

The previous MF shaping parameters are allocated to the FLC three variables e, Δe and Δu as follows:

- $SP_M e$, $SP_M \Delta e$ and $SP_M \Delta u$
- $SP_E e$, $SP_E \Delta e$ and $SP_E \Delta e$
- $SP_I e$, $SP_I \Delta e$ and $SP_I \Delta u$.

Note that if the medium and extreme MF shaping parameters are equal, all the UD MFs will have the same shape generated by the parameters value.

It is also important to prevent important overlapping between the generated MFs which is undesirable in fuzzy control (flattening phenomenon) [41]. For this purpose, we have fixed a maximum value to the space F_{SP} equal to the half of the minimal distance between the two nearby summits.

4.3.4 Parameter encoding

To run the SAA, suitable encoding for each of the optimizing parameters needs to be specified in terms of variation range, precision step and number of bits, since we use a binary encoding for a more thorough solution space exploration. Indeed, it is well known in control applications that it is recommended to use binary encoding to allow meticulous research by the metaheuristic

algorithms. After many tests, we have adopted the data given in Table 5.

5 Simulation Study

In order to validate the proposed FLC based control schemes, digital simulations have been carried out on the basis of the adopted discrete-time process transfer functions.

The simulation study has been conducted according to the basic feedback control system architecture shown in Figure 9, with three different web server workloads (A, B, C) for a better effectiveness and robustness evaluation.

5.1 FLC application

After long series of trial/error tests, the following characteristics have been fixed for the two cases of FLC based web server control; i.e. absolute service delay and relative service delay guarantees:

- Five FSs have been chosen to describe the error, its rate of change and control variation amplitudes. As seen above, their linguistic formulation and symbols are defined in the usual fuzzy logic terminology by: Positive Big (PB), Positive Medium (PM), Zero (ZE), Negative Medium (NM), Negative Big (NB). The "meaning" of each linguistic value should be clear from its mnemonic.
- The set of decision rules forming the "rule base" which characterizes our strategy to control the studied dynamic process is organized in a matrix form (see Table 6) based on Mac Vicar-Whelan's diagonal decision table [39].
- The same triangular shapes have been assigned to the MFs of the FLC variables with a uniform distribution and a 50% overlap has been provided for the neighboring FSs (see Figure 10). Therefore, at any given point of the UD, no more than two FSs will have non-zero degree of membership.
- Often, for greater flexibility in FLC design and tuning, the universes of discourse for each process variable are "normalized" to the interval [−1,+1] by means of constant scaling factors.
- The scaling factors best values have been determined by a tedious trial-and-error process (see Table 7).
- The adopted inference method is based on the Mamdani's Implication mechanism. It is also called *SUPremum-MINimum composition principle* [35].
- To obtain crisp values of the inferred fuzzy control actions, we have selected the *Centre-Of-Gravity* defuzzification technique [42] which is the most commonly employed.

The obtained results are shown in Figure 11.

The FLC used to enforce the absolute and RD succeed to make the system output converge to the desired delay in an acceptable delay and maintain it at the vicinity of the reference before and after the two changes of workload occurring at 10 s and 20 s respectively. However, at these instants, inevitable but minor overshoots and undershoots occur due to the workload burst variations. Nevertheless, the FLC shows rather good robustness in the face of these situations.

To try to improve the obtained performances, we have applied the SAA as a tuning procedure in designing an optimized FLC. The SAOFLC application to the studied control system, in the same conditions, is presented in next subsection.

5.2 SAOFLC application

The SAOFLC based feedback control system architecture is shown in Figure 12.

As described above, the SAA optimization process starts with a first FLC FC_0 as an initial solution and begins the iterative evaluation of the generated new solutions by an objective (cost) function *Of*.

Of is chosen to maximize the inverse of the well known and the most adopted performance index: Integral of Time-weighted Absolute Error (ITAE) [43] abbreviated, here, by DITAE for its discrete form.

The mathematical expression of *Of*, minimized by the SAA, can be written as:

$$Of = \frac{1}{DITAE} = \frac{1}{\sum_{m=m_0}^{m=m_f} [mT_s * |e(mT_s|]}$$

where:

- m_0 and m_f are the initial and final discrete times of the evaluating period

Table 5 Encoding parameters

Parameter	*NFS*	PSG_1	PSG_2	*Angle*	PSF_1	PSF_2	*SP*	G_e, $G_{\Delta e}$		$G_{\Delta u}$
								RD case	AD case	
Interval	[3,9]	[0.1,1]	[-1,1]	[0,π/2]	[0.1,1]	[-1,1]	[0,1.99]	[0.01,1]	[-1,-0.01] ∪ [0.01,1]	[0.1,1]
Precision	2	0.01	2	π/512	0.01	2	0.01	0.01	0.01	0.1
Number of encoding bits	2	7	1	9	7	1	8	7	8	4

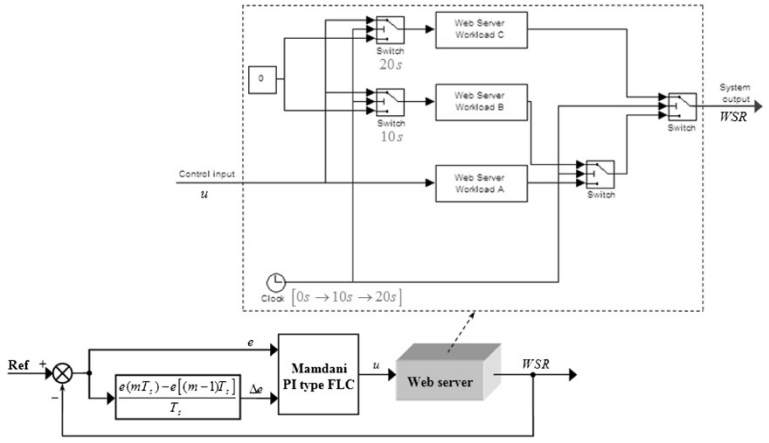


Figure 9 FLC based feedback control system architecture.

- T_s is the sampling period
- $e(mT_s) = W_k(mT_s) - C_k(mT_s)$ is the error, i.e., the difference, at a sampling instant, between the reference (set value) or the desired delay of class k (the desired delay ratio between class k and k-*1*) and the system response or the measured delay of class k (measured delay ratio between class k and k-*1*.

The algorithm for FLC optimal tuning based on the SA method is applied and the resulting controller parameters are set. As illustrated in Figure 12, red dashed lines are used to represent the representative signals of optimization.

During the search process, the SAA looks for the optimal setting of the FLC controller parameters which minimize the cost function *Of*. Solutions with low DITAE are considered as the fittest.

The SAA parameters chosen for the tuning purpose are shown in Table 8.

After the optimization process, the main characteristics (decision table, scaling factors and MFs) have been fixed for the two cases of FLC based web server control; i.e. absolute service delay and relative service delay guarantees as shown in Table 9, 10, Figures 13 and 14.

The digital simualtion results, illustrating the performances of the implemented SAOFLC applied to provide better QoS than those achieved by the classic Mamdani FLC, in the two considered cases (AD control and RD control) are shown in Figure 15 (a) and Figure 15 (b) respectively.

As can be seen from these figures, the optimized controller exhibits rather better step response performance in terms of rise time, overshoot magnitude, oscillations around the reference (desired delay difference (ratio)) and response (settling) time. We can also see that the SAOFLC shows an improvement in terms of robustness when faced to the simulated sudden workload variations (very hard task for the controller), particularly for the RD case.

Under the SAOFLC strategy, the closed-loop controlled web server enforces, succesfully, the absolute (relative) delay guarantee by satisfying the required delay difference (delay ratio) for the high priority classes (class 0 and class 1) with an obvious superiority than the standard Mamdani type FLC.

Table 6 5X5 Mc Vicar-Whelan decision table

Δ*u*		*e*				
		NB	**NM**	**ZE**	**PM**	**PB**
	NB	NB	NB	NB	NM	ZE
	NM	NB	NB	NM	ZE	PM
Δ*e*	**ZE**	NB	NM	ZE	PM	PB
	PM	NM	ZE	PM	PB	PB
	PB	ZE	PM	PB	PB	PB

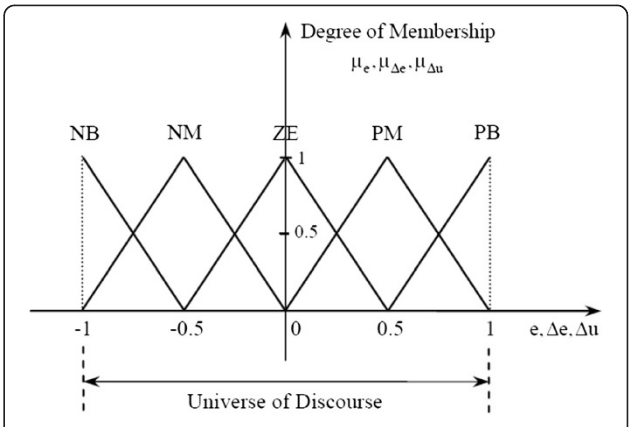


Figure 10 MFs for *e*, Δ*e* and Δ.

Table 7 FLC scaling factors

	G_e	$G_{\Delta e}$	$G_{\Delta u}$
AD control	0.4	3	0.01
RD control	0.29	1	0.012

6 Related work

The problem of QoS performance enhancement for Web servers is an attractive research field. Even though several works have extensively investigated different QoS enhancing mechanisms supporting service differentiation, few research works addressing the application of feedback control methodologies are available.

We start our description on literature review of related works by pointing out some pertinent research works that have employed service delay differentiation approaches as mechanisms of QoS enhancement. We have found very interesting the investigations of Leung et al. [44], Tham and Subramaniam [45], Lee et al. [46], Li et al. [47], Rashid et al. [48], Wei et al. [49], Bourasa and Sevasti [50], Wu et al. [51], Garcia et al. [52], Dimitriou and Tsaoussidis [53], Gao et al. [54], and Varela et al. [55].

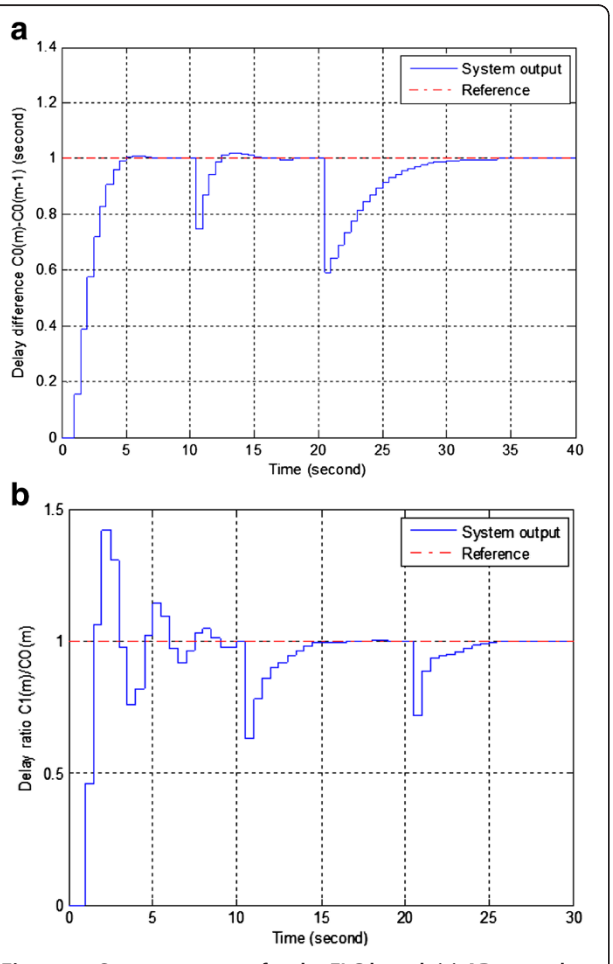


Figure 11 System response for the FLC based: (a) AD control (b) RD control for the three considered workloads.

The closest works to our investigation being those using feedback control techniques, we briefly present some relevant ones in a chronological order.

Andersson et al. [10] adopted a combination of queuing theory and control theory. The Apache web server has been modeled as a GI/G/1-system. Then, a standard PI-controller was employed as an admission control mechanism.

Henriksson et al. [56] presented a contribution as an extension of the classical combined feedforward/feedback control framework where the queuing theory is used for feedforward delay prediction. They replace the queuing model with a predictor that uses instantaneous measurements to predict future delays. The proposed strategy was evaluated in simulation and by experiments on an Apache web server.

Oottamakorn [57] proposed a resource management and scheduling algorithm to provide relative delays differentiated guarantees to classes of incoming requests at a QoS-aware web server. One of the key results of his work is the development of an efficient procedure for capturing the predictive traffic characteristics and performances by monitoring ongoing traffic arrivals. This allows the web server's resource management by determining sufficient server resource for each traffic class in order to meet its delay requirements. In order to achieve a self-stabilizing performance in delay QoS guarantees, he has implemented an adaptive feedback control mechanism.

The paper of Lu et al. [13] is the most important work upon which we have based our investigation. In this paper, the authors presented the design and implementation of an adaptive Web server architecture to provide relative and absolute connection delay guarantees for different service classes. Their first contribution is an adaptive architecture based on feedback control loops that enforce desired connection delays via dynamic connection scheduling and process reallocation. The second contribution is the use of control theoretic techniques (PI controllers based on the Root Locus method) to model and design the feedback loops with desired dynamic performance. Their adaptive architecture was implemented by modifying an Apache server.

Zhou et al. [15] investigated the problem of providing proportional QoS differentiation with respect to response time on Web servers. They first present a processing rate allocation scheme based on the foundations of queueing theory. They designed and implemented an adaptive process allocation approach, guided by the queueing-theoretical rate allocation scheme, on an Apache server. They established that this application-level implementation shows weak QoS predictability because

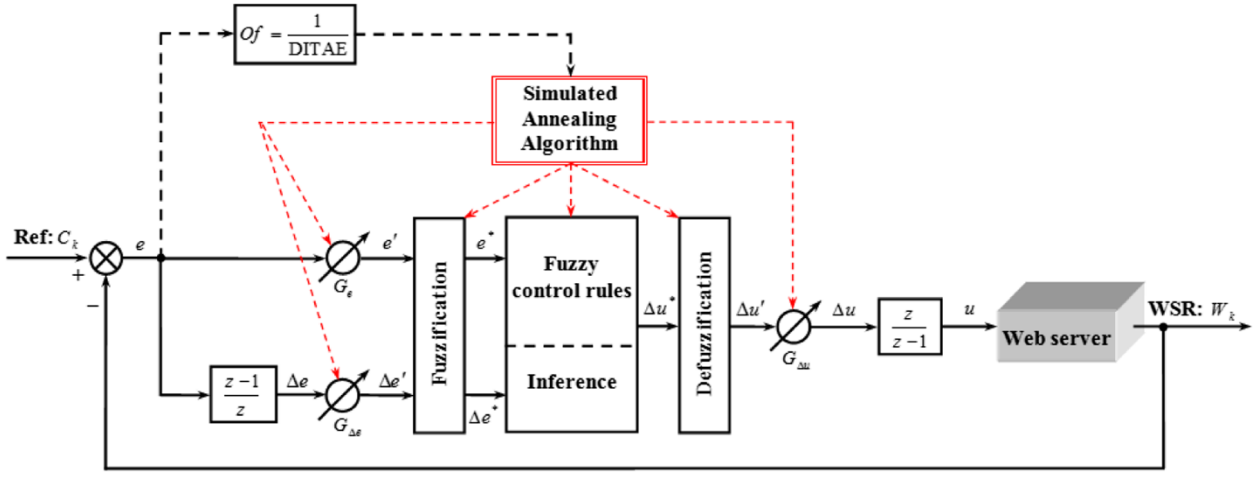


Figure 12 SAOFLC based feedback control system architecture.

it does not have fine-grained control over the consumption of resources that the kernel consumes and hence the processing rate is not strictly proportional to the number of processes allocated. They then designed a feedback controller and integrated it with the queueing-theoretical approach. The adopted feedback control strategy adjusts process allocations according to the difference between the target response time and the achieved response time using a Proportional-Integral -Derivative (PID) controller.

Qin and Wang [16] applied a control-theoretic approach to the performance management of Internet Web servers to meet service-level agreements. In particular, a CPU frequency management problem has been studied to provide response time guarantees with minimal energy cost. It was argued that linear time-invariant modeling and control may not be sufficient for the system to adapt to dynamically varying load conditions. Instead, they adopted a linear-parameter-varying (LPV) approach.

Kihl et al. [18] presented how admission control mechanisms can be designed with a combination of queuing theory and control theory. They modeled an Apache web server as a GI/G/1-system and validated their model as an accurate representation of the experimental system, in terms of average server utilization. Using simulations for discrete-event systems based on queuing theory and with experiments on an Apache web server, they compared a PI controller and an RST-controller, both commonly used in automatic control, with a static controller and a step controller, both commonly used in telecommunication systems. Note that the controllers were implemented as modules inside the Apache source code. They have also performed a nonlinear stability analysis for the PI-controlled system.

In Yansu et al. [19], a self-tuning control framework to provide proportional delay differentiation guarantees on Web Server has been proposed. The approach updates the model and controller parameters based on the variations of object model to reduce system error and optimize the performances through an online identification.

In Lu et al. (Lu J, Dai G, Mu D, Yu J, Li H [58] QoS Guarantee in Tomcat Web Server: A Feedback Control Approach. In: Proceedings of the 2011), the authors considered providing two types of QoS guarantees, proportional delay differentiation and absolute delay guarantee, in the database connection pool in Tomcat Web server

Table 8 Simulated annealing algorithm parameters

SA property	Method/value
Neighborhood generation method	swap of two elements
Initial temperature (T)	85
Final temperature (T_{fin})	3
Maximum number of iterations	100
Neighbor list size	30

Table 9 Decision table of the SAOFLC for the two cases

Δu		e						
		NB	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
Δe	**NVB**	NB	NB	NB	ZE	PB	PB	PB
	NB	NB	NB	NB	ZE	PB	PB	PB
	NM	NB	NB	NB	ZE	PB	PB	PB
	NS	NB	NB	NB	ZE	PB	PB	PB
	ZE	NB	NB	NB	ZE	PB	PB	PB
	PS	NB	NB	NB	ZE	PB	PB	PB
	PM	NB	NB	NB	ZE	PB	PB	PB
	PB	NB	NB	NB	ZE	PB	PB	PB
	PVB	NB	NB	NB	ZE	PB	PB	PB

Table 10 SAOFLC scaling factors

	G_e	$G_{\Delta e}$	$G_{\Delta u}$
AD control	0.7638	−0.0394	1
RD control	0.2381	0.6032	0.4286

application servers using the classical feedback control theory. To achieve these goals, they established approximate linear time-invariant models through system identification experimentally, and designed two PI controllers using the root locus method. These controllers are invoked periodically to calculate and adjust the probabilities for different classes of requests to use a limited number of database connections, according to the error between the measured QoS metric and the reference value.

In a recent work, Patikirikorala et al. [59] proposed a new approach for QoS performance management and resource provisioning by using an off-line identification of Hammerstein and Wiener nonlinear block structural model. Using the characteristic structure of the nonlinear model, a predictive feedback controller based on a gain schedule technique is incorporated in the design to achieve the performance objectives.

Examples of earlier research investigations using fuzzy logic based feedback control can be found in Diao et al. [9], Wei et al. [11], Chan and Chu [12], Wei et al. [14], Wei et al. [60], Tian et al. [20], Rao et al. [21].

In this paper, we have investigated the capabilities of two PI type Mamdani FLCs. The first has been obtained by trial-and-error process and the second synthesized by a SA based optimization.

Note that we have conducted performance evaluation of the proposed intelligent feedback control strategies based on validated mathematical models established by Lu et al. [13]. Our work focuses mainly on testing their robustness when faced with abrupt workload variations.

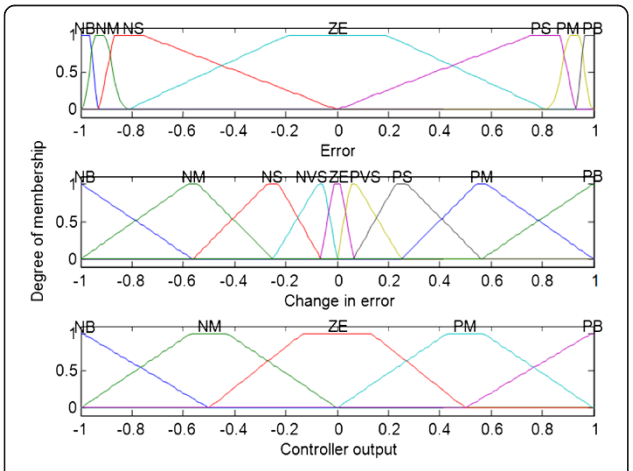


Figure 13 MFs of the SAOFLC: AD case.

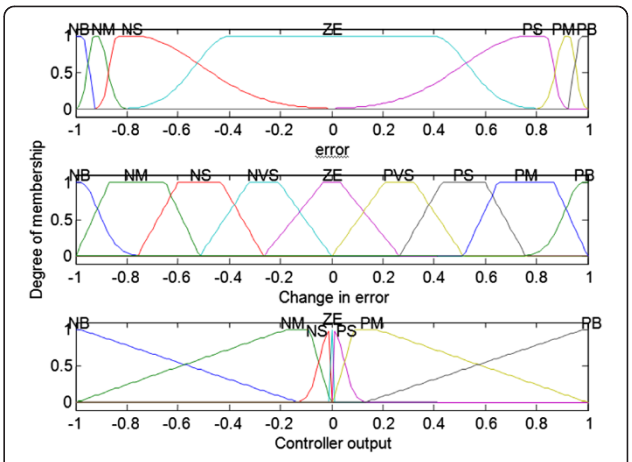


Figure 14 MFs of the SAOFLC: RD case.

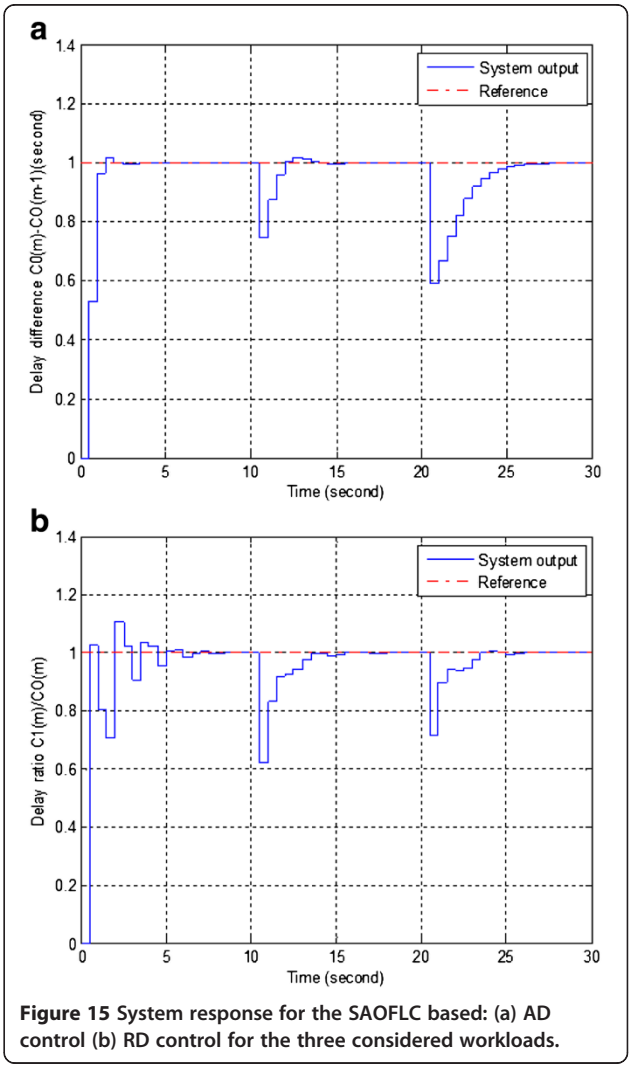


Figure 15 System response for the SAOFLC based: (a) AD control (b) RD control for the three considered workloads.

7 Conclusion and further work

This paper has addressed the QoS feedback intelligent control of a web server by considering its two common models in service differentiation: the absolute delay and the relative delay guarantees.

The application of two fuzzy logic controllers has been investigated as robust solutions for enforcing desired service performances in face of unpredictable server workloads: a Mamdani type fuzzy logic controller (FLC) and a simulated annealing optimized FLC (SAOFLC).

The main contributions of the proposed optimizing approach have been revealed in the tuning procedures of all the FLC design parameters through the minimization of a performance index. Explicitly, the innovations concern:

- the technique of fuzzy sets assignment to each of the grid nodes in the special case of equality of minimal distances between the points representing the candidate decision rules
- the formulation of the spacing law in function of the spacing parameter in the decision rules table deriving method
- the formulations linking the trapezoidal and the two-sided Gaussian membership functions
- the optimization and the diversification of the membership functions shapes offering possibilities of hybridization on the universe of discourse of each of the FLC input/output variables
- a simple solution to prevent important overlapping between the generated membership functions.
- The digital simulations have allowed us to validate the effectiveness of the proposed structures of control. Indeed, both of the FLC and the SAOFLC capabilities have been evaluated when applied to guarantee desired dynamic performance of the web server delay services.

Both of the adopted intelligent control strategies have realized quite satisfactory results. But, it has been clearly noted that the optimized FLC achieves rather high control performances in comparison with those of the standard Mamdani FLC in terms of transition and steady-state response characteristics.

Further studies to improve the obtained performances by other feedback control schemes as well as the optimization by other techniques such as tabu search, genetic algorithm, ant colonies, swarm techniques, bio-inspired techniques ... will be conducted as well.

Competing interest

The authors declare that they have no competing interest.

Authors's contributions

ML and YS created and developed the proposed approaches. SR participated in the experiments. ML and SR wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work was partially sponsored by MESRS/DGRSDT/CERIST/PNR8/E166/4884. We also would like to thank the anonymous reviewers who greatly contributed to the betterment of this work.

Received: 18 February 2012 Accepted: 27 February 2013
Published: 14 June 2013

References

1. Wang Z (2001) Internet QoS. Architectures and mechanisms for quality of service. Morgan Kaufmann, San Fransisco, CA, USA
2. Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W (1998) An architecture for differentiated services. IETF. Request for Comments 2475
3. Kilkki K (1999) Differentiated services for the internet. Macmillan Technical Publishing, Indianapolis, IN, USA
4. Abdelzaher TF, Stankovic JA, Lu C, Zhang R, Lu Y (2003) Feedback performance control in software services. IEEE Control Syst 23(3):74–90
5. Hellerstein JL, Diao Y, Parekh S, Tilbury DM (2004) Feedback control of computing systems. IEEE Press-Wiley, Hoboken, NJ, USA
6. Abdelzaher TF, Diao Y, Hellerstein JL, Lu C, Zhu X (2008) Introduction to control theory and its application to computing systems. In: Liu Z, Xia CH (ed) Performance Modeling and Engineering. Springer, pp 185–215. Part II, Chapter 7
7. Parekh S (2010) Feedback control techniques for performance management, Ph.D Dissertation. University of Washington, Seattle, WA, USA
8. Lu C, Abdelzaher TF, Stancovic JA, Son SH (2001) A feedback control approach for guaranteeing relative delays in web servers. Proccedings of the Seventh IEEE Real-Time Technology and Applications Symposium, Taipei, Taiwan, pp 51–62
9. Diao Y, Hellerstein JL, Parekh S (2002) Optimizing quality of service using fuzzy contro. In: Feridun M, Kropf P, Babon G (ed) Management Technologies for E-commerce an E-Business Applications. Lecture Notes in Computer Science, 2506th edition. Springer, Berlin, pp 42–53
10. Andersson M, Kihl M, Robertsson A (2003) Modelling and Design of Admission Control Mechanisms for Web Servers using Non-linear Control Theory. In: Proceedings of the ITCom's Conference on Performance and Control of Next-Generation Communication Networks. SPIE proceedings series, 5244th edition. , Orlando, FL, USA, pp 53–64
11. Wei Y, Lin C, Chu X, Shan Z, Ren F (2005) Class-Based Latency Assurances for Web Servers. In: High Performance Computing and Communications. Lecture Notes in Computer Science, 3726th edition. Springer, Berlin, pp 388–394
12. Chan KH, Chu X (2006) Design of a fuzzy PI controller to guarantee proportional delay differentiation on web servers. Technical Report COMP-06-001. Department of Computer Science, Hong Kong Baptist University
13. Lu C, Abdelzaher TF, Stancovic JA, Son SH (2006) Feedback control architecture and design methodology for service delay guarantees in web servers. IEEE Trans on Parallel Distrib Syst 17(9):1014–1027
14. Wei Y, Xu C-Z, Zhou X, Li Q (2006) Fuzzy control for guaranteeing absolute delays in web servers. Int J High Performance Comput Netw 4(5–6):338–346
15. Zhou X, Cai Y, Chow E (2006) An integrated approach with feedback control for robust web QoS design. Comput Commun 29(16):3158–3169
16. Qin W, Wang Q (2007) Modeling and control design for performance management of web servers via an LPV approach. IEEE Trans Contr Syst Tech 15(2):259–275
17. Pan W, Mu D, Wu H, Yao L (2008) Feedback control-based QoS guarantees in web application servers. In: Proceedings of the IEEE International Conference on High Performance Computing and Communications, Dalian, China, pp 328–334
18. Kihl M, Robertsson A, Andersson M, Wittenmark B (2008) Control-theoretic Analysis of Admission Control Mechanisms for Web Server Systems. World Wide Web 11(1):193–116
19. Yansu H, Guanzhong D, Ang G, Wenping P (2009) A self-tuning control for web QoS. In: Proceedings of the International Conference on Information Engineering and Computer Science, Wuhan, China, pp 1–4

20. Tian F, Xu W, Sun J (2010) Web QoS control using fuzzy adaptive PI controller. Proceedings of the International Symposium on Distributed Computing and Applications to Business Engineering and Science, Hong Kong, pp 72–75
21. Rao J, Wei Y, Gong J, Xu C-Z (2011) DynaQoS: model-free self-tuning fuzzy control of virtualized resources for QoS provisioning. In: Proceedings of the 19th International Workshop on Quality of Service (IWQoS'11). IEEE Press, San Jose, CA, USA, pp 1–9
22. Venkatarama HS, Sekaran KC (2012) Autonomic Computing: A Fuzzy Control Approach towards Application Development. In: Cong-Vinh P (ed) Formal and Practical Aspects of Autonomic Computing and Networking: Specification, Development, and Verification. IGI Global, Hershey, PA, USA, pp 118–134. Chapter 5
23. Lama P, Zhou X (2012) Efficient Server Provisioning with Control for End-to-End Response Time Guarantee on Multitier. IEEE Trans on Parallel and Distributed Systems 23(1):78–86
24. Gourley D, Totty B, Sayer M, Aggarwal A, Reddy S (2002) HTTP: The Definitive Guide, O'Reilly Media
25. Kozierok CM (2005) The TCP/IP Guide: A Comprehensive. No Starch Press, Illustrated Internet Protocols Reference
26. Andersson M (2005) Introduction to Web Server Modeling and Control Research. Technical Report, Department of Communication Systems, Lund Institute of Technology
27. Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P, Berners-Lee T (1999) Hypertext Transfer Protocol-HTTP/1.1. IETF RFC 2616
28. http://news.netcraft.com/archives/2012/02/07/february-2012-web-server-survey.html
29. Lee CC (1990) Fuzzy logic in control systems: fuzzy logic controller- part I & part II. IEEE Trans on Systems Man and Cybernetics 20(2):404–435
30. Mamdani EH (1974) Applications of fuzzy algorithms for control of a simple dynamic plant. Proceedings of the IEE 121(12):1585–1588
31. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087–1092
32. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
33. Ljung L (1999) System Identification - Theory For the User, 2nd edition. PTR Prentice Hall, Upper Saddle River, N.J., USA
34. Barford P, Crovella ME (1998) Generating Representative Web Workloads for Network and Server Performance Evaluation. Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, Madison, WI, USA, pp 151–160
35. Park YJ, Cho HS, Cha DH (1995) Genetic algorithm-based optimization of fuzzy logic controller using characteristic parameters. Proceedings of the IEEE International Conference on Evolutionary Computation, Perth, WA, Australia, pp 831–836
36. Foran J (2002) Optimisation of a fuzzy logic controller using genetic algorithms. Master of Engineering Project Report. Dublin City University, School of Electronic Engineering
37. Loudini M (2007) Contribution à la modélisation et à la commande intelligente d'un bras de robot manipulateur flexible. Ph.D. thesis, Electrical Engineering Dept., Ecole Nationale Polytechnique, Algiers, Algeria
38. Illoul R, Loudini M, Selatnia A (2011) Particle swarm optimization of a fuzzy regulator for an absorption packed column. Mediterranean Journal of Measurement and Control 7(1):174–182
39. Mac Vicar-Whelan PJ (1976) Fuzzy sets for man machine interactions. Int J of Man–machine Studies 8(6):687–697
40. Cheong F, Lai R (2000) Constraining the optimization of a fuzzy logic controller using an enhanced genetic algorithm. IEEE Trans Syst Man Cybern B Cybern 30(1):31–46
41. Bühler H (1994) Réglage par logique floue. Presses Polytechniques et Universitaires Romandes. Lausanne, Switzerland
42. Jager R, Verbruggen HB, Bruijn PM (1992) The role of defuzzification methods in the application of fuzzy control. Proceedings of the IFAC Symposium on Intelligent Components and Instuments for Control Applications, Malaga, Spain, pp 75–80
43. Graham D, Lathrop RC (1953) The synthesis of optimum transient response: Criteria and standard forms. Transacactions of the American Institute of Electrical Engineers, Applications and Industry 72:273–288
44. Leung MKH, Lui JCS, Yau DKY (2001) Adaptive proportional delay differentiated services: characterization and performance evaluation. IEEE/ACM Transactions on Networking 9(6):80–817
45. Tham C-K, Subramaniam VR (2002) Integrating web server and network QoS to provide end-to-end service differentiation. In: Proceedings of the 10th IEEE International Conference on Networks (ICON 2002). , Singapore, pp 389–394
46. Lee SCM, Lui JCS, Yau DKY (2004) A proportional-delay DiffServ-enabled Web server: admission control and dynamic adaptation. IEEE Trans Parallel Distrib Syst 15(5):385–400
47. Li ZG, Chen C, Soh YC (2004) Relative differentiated delay service: time varying deficit round robin. Proceedings of the Fifth World Congress on Intelligent Control and Automation, Hangzhou, China, pp 5608–5612
48. Rashid MM, Alfa AS, Hossain E, Maheswaran M (2005) An analytical approach to providing controllable differentiated quality of service in web servers. IEEE Trans Parallel Distrib Syst 16(11):1022–1033
49. Wei J, Xu C-Z, Zhou X, Li Q (2006) A robust packet scheduling algorithm for proportional delay differentiation services. Comput Commun 29(18):3679–3690
50. Bourasa C, Sevasti A (2007) An analytical QoS service model for delay-based differentiation. Computer Networks 51(12):3549–3563
51. Wu C-C, Wu H-M, Lin W (2008) High-performance packet scheduling to provide relative delay differentiation in future high-speed networks. Comput Commun 31(10):1865–1876
52. Garcia DF, Garcia J, Entrialgo J, Garcia M, Valledor P, Garcia R, Campos AM (2009) A QoS control mechanism to provide service differentiation and overload protection to internet scalable servers. IEEE Trans on Services Computing 2(1):3–16
53. Dimitriou S, Tsaoussidis V (2010) Promoting effective service differentiation with Size-oriented Queue Managemen. Computer Networks 54(18):3360–3372
54. Gao A, Mu D, Hu Y (2011) A QoS control approach in differentiated web cashing service. J of Networks 6(1):62–70
55. Varela A, Vazão T, Arroz G (2012) Providing service differentiation in pure IP-based networks. Comput Commun 35(1):33–46
56. Henriksson D, Lu Y, Abdelzaher T (2004) Improved prediction for web server delay control. In: Proceedings of the 16th Euromicro Conference on Real-Time Systems. IEEE Computer Press, Catania, Sicily, Italy, pp 61–68
57. Oottamakorn C (2005) Class-based guarantees of relative delay services in web servers. In: Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2005). part of the 23rd Multi-Conference on Applied Informatics, Innsbruck, Austria, pp 417–423
58. Lu J, Dai G, Mu D, Yu J, Li H (2011) QoS Guarantee in Tomcat Web Server: A Feedback Control Approach. In: Proceedings of the (2011) International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Beijing, China, pp 183–189
59. Patikirikorala T, Wang L, Colman A, Han J (2012) Hammerstein–Wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems. Control Eng Pract 20(1):49–61
60. Wei J, Xu CZ (2007) Consistent proportional delay differentiation: A fuzzy control approach. Computer Networks 51(5–6):2015–2032

doi:10.1186/1869-0238-4-15
Cite this article as: Loudini *et al.*: **Incorporate intelligence into the differentiated services strategies of a Web server: an advanced feedback control approach.** *Journal of Internet Services and Applications* 2013 **4**:15.